Chapter 1

PBS PRO: GRID COMPUTING AND SCHEDULING ATTRIBUTES

Bill Nitzberg,[1] Jennifer M. Schopf,[2] and James Patton Jones[1]
[1] *Altair Grid Technologies*
[2] *Mathematics and Computer Science Division, Argonne National Laboratory*

Abstract The PBS Pro software is a full-featured workload management and job scheduling system with capabilities that cover the entire Grid computing space: security, information, compute, and data. The security infrastructure includes user authentication, access control lists, X.509 certificate support, and cross-site user mapping facilities. Detailed status and usage information is maintained and available both programmatically and via a graphical interface. Compute Grids can be built to support advance reservations, harvest idle desktop compute cycles, and peer schedule work (automatically moving jobs across the room or across the globe). Data management in PBS Pro is handled via automatic stage-in and stage-out of files. The PBS Pro system has numerous site-tunable parameters and can provide access to available scheduling information, information about requesting resources, allocation properties, and information about how an allocation execution can be manipulated.

1. INTRODUCTION

The Portable Batch System, Professional Edition (PBS Pro), is a flexible workload management and batch job scheduling system originally developed to manage aerospace computing resources at NASA. PBS Pro addresses issues of resource utilization in computing-intense industries and forms the basis of many Grid computing projects.

The PBS Pro software includes capabilities that cover the entire Grid computing space: security, information, compute, and data. We look at the Grid capabilities of PBS Pro 5.3 circa March 2003, as well as how they relate to the scheduling attributes detailed in Chapter **??**.

2. HISTORY

PBS has been used in the areas of workload management and Grid computing over the past decade. In the early 1990s NASA needed to replace its outdated NQS batch system but found nothing suitable on the market. Hence NASA led an international effort to generate a list of requirements for a next-generation resource management system. The requirements and functional specification were soon adopted as an IEEE POSIX standard [IEE94]. Next, NASA funded (via the R&D contractor MRJ/Veridian) the design and development of a new resource management system compliant with the standard. Thus, in 1993 the Portable Batch System was born [Hen95]. PBS began to be used on distributed parallel systems and replaced NQS on traditional supercomputers and server systems. Eventually the industry evolved toward distributed parallel systems, taking the form of both special-purpose and commodity clusters. The PBS story continued when Veridian released the professional edition of PBS (PBS Pro), an enterprise-quality workload management solution. Most recently, in January 2003, the PBS technology and associated engineering team were acquired by Altair Engineering, Inc., and set up as a separate, subsidiary company (Altair Grid Technologies) focused on continued development of the PBS Pro product line and Grid computing.

In the mid-1990s PBS was selected as the enabling software for Grid computing (then called *metacomputing*). Examples such as the NASA Metacenter (1996-1997 [Jon96, Jon97a]), the Department of Defense Meta-queueing Project (1997-1998 [Jon97b, Jon98]), and NASA's Information Power Grid (1998-2003+ [JGN99]) demonstrate this capability. As a founding participant of the Global Grid Forum (GGF, see also [GGF]), and co-director of the GGF Scheduling Area, the PBS Pro team has committed to furthering Grid computing technologies.

3. GRID CAPABILITIES

Workload management software such as PBS Pro is a key component of Grid computing. It is middleware technology that sits between compute-intensive or data-intensive applications and the network, hardware, and operating system. The software aggregates all the computing and data resources into a single virtual pool. It schedules and distributes all types of application runs (serial, parallel, distributed memory, etc.) on all types of hardware (desktops, clusters, and supercomputers and even across sites) with selectable levels of security. An overview of the basic Grid capabilities (security, information, compute, and data) is provided in this section. Details of features can be found in the PBS Pro documentation [Jon03a, Jon03b].

3.1 Security

Perhaps the most fundamental capabilities of Grid infrastructure are secure authentication (proving one's identity) and authorization (granting permission). The security capabilities of PBS Pro cover both user and host authentication as well as authorization.

Internally, authentication and authorization are user name based (UNIX or Windows login). Authentication uses standard UNIX and Windows security (with additional access checks based on stringent hostname-IP address rules). Authorization is handled by complex access control lists (ACLs), which permit access restriction (or permission) via user, group, system, and network.

X.509 certificates, the *de facto* Grid standard for identification and authentication, are also supported. PBS Pro can pick up the user's certificate at job submission and automatically create a proxy on the execution nodes assigned to that user's job. The distinguished name (DN) from the certificate is carried with the job throughout its lifetime and is written to the PBS accounting logs as part of the job accounting record. If a user's certificate expires, PBS Pro will place the job on hold and notify the user to renew the certificate.

Furthermore, user identity mapping between sites is handled by a mapping function (and can be set up similarly to the gridmap file used as part of the Globus Toolkit [FK97, GLO]).

3.2 Information

If security is the first fundamental capability of Grid infrastructure, then information management is a close second. Access to the state of the infrastructure itself (e.g., available systems, queue lengths, software license locations), is required to support automatic aggregation of Grid components as well as optimizing assignment of resources to Grid activities. PBS Pro monitors both resource state and common workload management information.

The PBS Pro system monitor and job executor daemon processes (MOMs) collect real-time data on the state of systems and executing jobs. This data, combined with less dynamic information on queued jobs, accounting logs, and static configuration information, gives a complete view of resources being managed. PBS protects this information with ACLs, which allow the PBS manager to ensure that, for example, only the owner of a job can view its current status. The node-utilization data collected by the PBS MOMs can be viewed graphically by using the `xpbsmon` command. Specifically, current node availability and status, node CPU and memory utilization, and assigned jobs are displayed by default. Other indices may be selected by the user.

This data can easily be integrated with larger Grid infrastructure databases (such as the information services within the Globus Toolkit). For example, NASA's Information Power Grid [JGN99] both pushes and pulls

data from PBS Pro into the Globus Toolkit Monitoring and Discovery Service (MDS2) [CFFK01, MDS].

3.3 Compute

In addition to traditional workload management capabilities, specific features of PBS Pro address the compute aspects of Grids. These include *advance reservation* support, *cycle harvesting*, and *peer scheduling*.

An *advance reservation* is a set of resources with availability limited to a specific user (or group of users), a specific start time, and a specified duration. Advance reservations can be used to support co-scheduling, especially among diverse types of Grid resources, for example, one can reserve all resources necessary for tomorrow's vehicle crash test experiment: computer cycles, network bandwidth, crash test database access, visualization systems, and the crash test facility itself.

Advance reservations are implemented in PBS Pro by a user (or a higher-level Grid scheduler) submitting a reservation with the `pbs_rsub` command (or API function). PBS Pro then checks to see whether the reservation conflicts with currently running jobs, other confirmed reservations, and dedicated time. A reservation request that fails this check is denied by the scheduler. Once the scheduler has confirmed the reservation, a queue is created to represent the reservation. The queue has a user-level access control list set to the user who submitted the reservation (or as specified by the higher-level scheduler) and any other users the owner specified. The queue then accepts jobs in the same manner as normal queues. When the reservation start time is reached, the queue is started. Once the reservation is complete, any jobs remaining in the queue or still running are deleted, and the reservation is removed from the server.

Cycle harvesting of idle workstations is a method of expanding the available computing resources by automatically including unused workstations that otherwise would be idle. This is particularly useful for sites that have a significant number of workstations that are unused during nights and weekends (or even during lunch). With this feature, when the *owner* of the workstation isn't using it, the machine can be configured to run PBS Pro jobs. If a system is configured for cycle harvesting, it becomes available for batch usage by PBS Pro if its keyboard and mouse remain unused or idle for a certain period of time, or if the system load drops below a site-configurable threshold (i.e., the workstation is shown to be in state *free* when the status of the node is queried). If the keyboard or mouse is used, the workstation becomes unavailable for batch work; PBS Pro suspends any running jobs on that workstation and does not attempt to schedule any additional work on it until the state changes.

Peer scheduling is a PBS Pro feature that enables a site (or multiple sites) to have different PBS Pro installations automatically run jobs from each other's queues. This provides the ability to dynamically load-balance across multiple, separate PBS Pro installations. These cooperating PBS Pro installations are referred to as *Peers*, and the environment is a peer-to-peer computational Grid environment. When peer scheduling is enabled and resources are available, PBS Pro can *pull* jobs from one or more (remote) *peer servers* and run them locally. No job will be moved if it cannot run immediately. When the scheduler determines that a remote job can run locally, it will move the job to the specified queue on the local server and then run the job. Since the scheduler maps the remote jobs to a local queue, any moved jobs are subject to the policies of the queue they are moved into. If remote jobs are to be treated differently from local jobs, this can be done on the queue level. A queue can be created exclusively for remote jobs, and this will allow queue-level policy to be set for remote jobs. For example, one can set a priority value on one's queues and enable sorting by priority to ensure that remotely queued jobs are always lower (or higher) priority than locally queued jobs.

3.4 Data

PBS Pro has long supported the most basic capability for implementing a data Grid: file staging. Users of PBS Pro can specify any number of input and output files needed by their application at job submission time. The PBS Pro system automatically handles copying files onto execution nodes (stage-in) prior to running the job, and copying files off execution nodes (state-out) after the job completes. PBS Pro will not run a job until all the files requested to be staged-in have successfully been copied. Multiple transport mechanisms are offered, including rcp, scp, and GridFTP [ABB+02].

The file staging feature of PBS Pro also supports the Globus Toolkit's Global Access to Secondary Storage (GASS) software. Given a complete stage-in directive, PBS Pro will take care of copying the specified input file over to the executing Globus Toolkit machine. The same process is used for a stage-out directive. Globus mechanisms are used for transferring files to hosts that run Globus; otherwise, the normal PBS Pro file transport mechanism is used.

3.5 Other Grid-Related Capabilities

Other Grid-related capabilities of PBS Pro include interfaces to Grid computing environments such as the Globus Toolkit [FK97, GLO] and UNICORE [UNI].

For example, PBS Pro can serve as a *front end* to Globus, permitting the user to submit jobs requesting Globus resources using the normal PBS Pro commands. When such a job is received, PBS Pro will translate the requests

into a Globus job, and then submit it to the requested site. In addition, PBS Pro can serve as a *back-end* to Globus, receiving jobs from Globus and running them according to local policy.

PBS Pro also acts as a *back end* system for the UNICORE Grid environment. Thus computational Grids built on UNICORE (such as the European DataGrid project [EUR]) can (and do) use PBS Pro as the underlying batch system.

4. ATTRIBUTE BY ATTRIBUTE ASSESSMENT

To help compare scheduling systems, in this section we detail PBS's approach using the attributes defined in Chapter **??**. These attributes were defined to aid in characterizing the features of a local resource management system that can be exploited by a Grid environment. They are grouped into four categories: access to available scheduling information, described in Section 4.1; information about requesting resources, described in Section 4.2; allocation properties, discussed in Section 4.3; and information about how an allocation execution can be manipulated, described in Section 4.4.

4.1 Access to Available Scheduling Information

The most basic information a local resource management system can express is the status of the current resource usage, and the upcoming schedule of the jobs.

In PBS Pro, any user can access basic information about the queues and their status using the `qstat` command. This provides a straightforward interface to basic data about the progress a job is making in the queue, and users can extrapolate possible starting times from this data. In addition, administrators of a PBS Pro installation have access to more detailed information about the order in which jobs will be run according to the site-defined scheduling policy.

When jobs are submitted to PBS Pro, an email address is specified for event notification. The user may specify that email be sent to this address when the job starts, ends, or aborts (the default if not specified). Alternatively, the user may request no email notification be performed at all.

4.2 Requesting Resources

Local scheduling systems differ not only in terms of the functionality they provide but also in the type of data used to request resources. The attributes in this category include data about allocation offers, cost information, advance reservation data, de-allocation information, co-scheduling data, and job dependencies allowed.

PBS generates a single resource solution to a *run my job* request, whether it is a standard batch job request or a request for an advance reservation (see

Section 4.3). Once a job is submitted, it will run unless a user cancels it, the system goes down, or it is preempted (see Section 4.4).

Whether a request to run a job includes an estimated completion time from the requestor is configurable. If this data is not included, however, getting a high utilization of the resources is extremely difficult. For example, the backfilling feature needs this data. this functionality is not a requirement of the PBS Pro infrastructure, and it can be configured differently for different queues.

PBS Pro allows the consideration of job dependencies as a part of the job submission process in a variety of flavors. It is simple for a user to specify a number of options, including: Run job Y after job X completes; If job X succeeds, run Job Y, otherwise run job Z; Run Y after X whether X completes or not; Run job Y only after a certain time; Run job X anytime after a specified time.

PBS Pro allows co-scheduling by simply configuring the queues of the system. Hence, a site can have the added benefit of co-scheduling not as a special case but as the norm, so extensive debugging of a rarely used allocation process isn't needed.

4.3 Allocation Properties

Different local resource management systems allow different flexibility with respect to how an allocation is handled. This includes whether an allocation can be revoked, what guarantees are made with respect to completion times, start attempts, and finish times, and whether allocations can change during a run.

In PBS Pro, the user (i.e., the allocation requestor) or an administrator can revoke any allocation (using the `qdel` and `pbs_rdel` commands), both while the job is queued and while the job is running. Jobs can also be preempted by the scheduler, as discussed in the next section, as a configurable option. Depending on the configuration, a preempted job can be suspended, checkpointed, requeued to start over, or terminated. In all cases, preemption implies at least a temporary revocation of the allocation.

In terms of a guaranteed completion time for an allocation, if a request is made for two hours of resource and the job starts at 1 pm, it will finish by 3 pm. The flip side to this situation (e.g., can a user specify that a two-hour job finishes by 3 pm?) can currently be done by using an advance reservation. Work is under development to allow this capability under normal job submissions.

At setup time, one can configure how many job completion attempts should be allowed. This information is needed because some tasks, such as data transfers, may not succeed on the first try. One can also configure whether an allocation is exclusive or not, meaning whether the resource is space-shared or time-shared. Advance reservations are allowed only on space-shared resources.

PBS Pro currently does not support a malleable allocation, that is, an allocation that allows the addition or removal of resources during run time. When this feature is supported by common MPI-2 implementations, it will be added to PBS Pro.

4.4 Manipulating the Allocation Execution

It can be beneficial for a local resource management system to modify a running allocation in order to better coordinate the execution of a set of jobs. Such modifications can be done by using preemption, checkpointing, migration, and restart.

PBS Pro provides a number of options for manipulating allocation executions. Any job can be requeued or restarted. Preemption is a configurable option for any resource, and a site can use a variety of algorithms to specify which jobs should get preempted, how often, and for how long. When preempted, a job can be checkpointed if the underlying operating system allows this, for example SGI Irix and Cray UNICOS. If jobs are checkpointed by a user, they can be requeued to start at the stage in that checkpoint file. Migration usually can be done on-the-fly, but not for MPI jobs as this feature is currently not supported within MPI implementations.

5. SUMMARY AND FUTURE

The Grid computing field is quite young, and only the most basic promise of Grid computing is available today. Although PBS Pro features cover the Grid computing space (security, information, compute, and data), capabilities are constantly being added and refined. PBS Pro development plans include extending the support for X.509 certificates, expanding data Grid support to actively manage network bandwidth, and continuing to drive Grid standards via the Global Grid Forum. In particular, the PBS Pro team is actively involved defining numerous Grid standards: DRMMA [DRM], OGSA [OGS], GRAAP [GRA], and UR [UR].

Acknowledgments

We thank the many people who have been involved with the PBS software throughout the years, especially Bob Henderson, who has led the core development team from the very beginning. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, Office of Science, under contract W-31-109-Eng-38.

References

[ABB+02] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Data management and transfer in high-performance computational Grid environments. *Parallel Computing Journal*, 28(5):749–771, 2002.

[CFFK01] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10)*, August 2001.

[DRM] GGF Distributed Resource Management Application API Working Group (DRMAA-WG). `http://www.drmaa.org/`.

[EUR] Eurogrid. `http://www.eurogrid.org`.

[FK97] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11(2):115–129, 1997.

[GGF] Global Grid Forum (GGF). `http://www.ggf.org`.

[GLO] Globus Project. `http://www.globus.org`.

[GRA] GGF Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG). `http://www.fz-juelich.de/zam/RD/coop/ggf/graap/graap-wg.html`.

[Hen95] Robert L. Henderson. Job scheduling under the Portable Batch System. In D. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing (Proceedings of the First International JSSPP Workshop; LNCS #949)*, pages 178–186. Springer-Verlag, 1995.

[IEE94] IEEE. *IEEE Standard for Information Technology, POSIX 1003.2D*. IEEE, 1994.

[JGN99] William E. Johnston, Dennis Gannon, and Bill Nitzberg. Grids as production computing environments: The engineering aspects of NASA's Information Power Grid. In *Proceedings of the Eighth IEEE International Symposium on High-Performance Distributed Computing (HPDC-8)*, 1999.

[Jon96] James Patton Jones. The NASA Metacenter. In *Proceedings of the NASA High Performance Computing and Communications Program / Computational Aerosciences Workshop (HPCCP/CAS)*, August 1996.

[Jon97a] James Patton Jones. Implementation of the NASA Metacenter: Phase 1 report. Technical Report NAS-97-027, NASA Ames Research Center, October 1997.

[Jon97b] James Patton Jones. PBS technology transfer to Department of Defense sites. Technical Report NASA Ames Quarterly Report, NASA Ames Research Center, October 1997.

[Jon98] James Patton Jones. Designing a metacenter: Recommendations to DoD MSRC ASC and CEWES. Technical Report Technology Transfer Whitepaper, NASA Ames Research Center, March 1998.

[Jon03a] James Patton Jones, editor. *PBS Pro 5.3 Administrator Guide*. Altair Grid Technologies, 2003.

[Jon03b] James Patton Jones, editor. *PBS Pro 5.3 User Guide*. Altair Grid Technologies, 2003.

[MDS] Globus Monitoring and Discovery System (MDS2). `http://www.globus.org/mds`.

[OGS] GGF Open Grid Services Architecture Working Group (OGSA-WG). `http://www.ggf.org/ogsa-wg/`.

[UNI] Unicore. `http://www.unicore.org/`.

[UR] GGF Usage Record Working Group (UR-WG). `http://www.gridforum.org/3_SRM/ur.htm`.